

Hacking Articles

Raj Chandel's Blog



Menu

[Home](#) » [Website Hacking](#) » [A Detailed Guide on httpx](#)

Website Hacking

A Detailed Guide on httpx

March 14, 2022 By Raj Chandel

Introduction

httpx is a fast web application reconnaissance tool coded in go by www.projectidscovey.io. With a plethora of multiple modules effective in manipulating HTTP requests and filtering out responses, it is proving to be an effective tool in Bug Bounty Hunter's arsenal. While tools like curl already exist that can perform almost all the features covered in this tool, httpx has its own place among the analysts because of its speed and ease of access. You can download the source code from [here](#).

Table of content

- Installation of go version 1.17
- Installation of httpx
- Basic usage
- Subdomain enum using subfinder and scanac
- Content probes
- Content comparers
- Content filters
- Rates and timeouts
- Show responses and requests

- Filtering for SQL injections
- Filtering for XSS reflections
- Web page fuzzing
- File output
- TCP/IP customizations
- Post login
- HTTP methods probe
- Routing through proxy
- Conclusion

Installation of go version 1.17

Installation and proper running of httpx tool depends on go version 1.17. You can download, extract, add go in environment variables as follows. I am using Kali on amd64 architecture. Please feel free to download the appropriate package for your system on go.dev/dl

```
1. wget https://go.dev/dl/go1.17.8.linux-amd64.tar.gz
2. tar -C /usr/local/ -xzf go1.17.8.linux-amd64.tar.gz
```

Please make sure that you add the following lines in ~/.zshrc file:

```
1. #go variables
2. export GOPATH=/root/go-workspace
3. export GOROOT=/usr/local/go
4. PATH=$PATH:$GOROOT/bin/:$GOPATH/bin
```

After you have added the lines, zshrc file can be loaded with the source command and then we'll be ready to go. If all goes well, the "go version" command will give version 1.17.8 as output.

```
1. source ~/.zshrc
2. go version
```

```

(root@kali)-[~]
# wget https://go.dev/dl/go1.17.8.linux-amd64.tar.gz
--2022-03-12 23:32:07-- https://go.dev/dl/go1.17.8.linux-amd64.tar.gz
Resolving go.dev (go.dev)... 216.239.34.21, 216.239.32.21, 216.239.36.21, ...
Connecting to go.dev (go.dev)|216.239.34.21|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://dl.google.com/go/go1.17.8.linux-amd64.tar.gz [following]
--2022-03-12 23:32:08-- https://dl.google.com/go/go1.17.8.linux-amd64.tar.gz
Resolving dl.google.com (dl.google.com)... 142.250.194.238, 2404:6800:4002:825::200e
Connecting to dl.google.com (dl.google.com)|142.250.194.238|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 134902354 (129M) [application/x-gzip]
Saving to: 'go1.17.8.linux-amd64.tar.gz'

go1.17.8.linux-amd64.tar.gz 100%[=====>] 128.65M 10.0MB/s in 14s

2022-03-12 23:32:23 (8.87 MB/s) - 'go1.17.8.linux-amd64.tar.gz' saved [134902354/134902354]

(root@kali)-[~]
# tar -C /usr/local/ -xzf go1.17.8.linux-amd64.tar.gz

(root@kali)-[~]
# nano ~/.zshrc

(root@kali)-[~]
# tail ~/.zshrc

# enable command-not-found if installed
if [ -f /etc/zsh_command_not_found ]; then
    . /etc/zsh_command_not_found
fi

#go variables
export GOPATH=/root/go-workspace
export GOROOT=/usr/local/go
PATH=$PATH:$GOROOT/bin:$GOPATH/bin

(root@kali)-[~]
# source ~/.zshrc

(root@kali)-[~]
# go version
go version go1.17.8 linux/amd64

```

Installation of httpx

Installation of the tool is also possible by cloning the github repository and using a makefile to compile but we have an easier alternative. We can use go install to do the same like:

```
1. go install -v github.com/projectdiscovery/httpx/cmd/httpx@latest
```

```
(root@kali)-[~]
# go install -v github.com/projectdiscovery/httpx/cmd/httpx@latest
go: downloading github.com/projectdiscovery/httpx v1.2.0
go: downloading github.com/projectdiscovery/gologger v1.1.4
go: downloading github.com/bluele/gcache v0.0.2
go: downloading github.com/logrusorgru/aurora v2.0.3+incompatible
go: downloading github.com/pkg/errors v0.9.1
go: downloading github.com/projectdiscovery/clistats v0.0.8
go: downloading github.com/projectdiscovery/cryptoutil v0.0.0-20210805184155-b5d2512f9345
go: downloading github.com/projectdiscovery/fdmax v0.0.3
go: downloading github.com/projectdiscovery/fileutil v0.0.0-20210926044607-04f32490aa21
go: downloading github.com/projectdiscovery/goconfig v0.0.0-20210804090219-f893ccd0c69c
go: downloading github.com/projectdiscovery/goflags v0.0.7
go: downloading github.com/projectdiscovery/hmap v0.0.2-0.20210917080408-0fd7bd286bfa
go: downloading github.com/projectdiscovery/httputil v0.0.0-20210816170244-86fd46bc09f5
go: downloading github.com/projectdiscovery/iputil v0.0.0-20210804143329-3a30fcde43f3
go: downloading github.com/projectdiscovery/mapcidr v0.0.8
go: downloading github.com/projectdiscovery/rawhttp v0.0.8-0.20210814181734-56cca67b6e7e
go: downloading github.com/projectdiscovery/retryablehttp-go v1.0.2
go: downloading github.com/projectdiscovery/stringsutil v0.0.0-20210830151154-f567170afdd9
go: downloading github.com/projectdiscovery/urlutil v0.0.0-20210805190935-3d83726391c1
go: downloading github.com/projectdiscovery/wappalyzergo v0.0.30
go: downloading github.com/remeh/sizedwaitgroup v1.0.0
go: downloading go.uber.org/ratelimit v0.2.0
go: downloading go.uber.org/atomic v1.9.0
go: downloading github.com/karrick/godirwalk v1.16.1
go: downloading github.com/json-iterator/go v1.1.12
go: downloading github.com/projectdiscovery/reflectutil v0.0.0-20210804085554-4d90952bf92f
go: downloading gopkg.in/ini.v1 v1.62.0
go: downloading github.com/cnf/structhash v0.0.0-20201127153200-e1b16c1ebc08
go: downloading gopkg.in/yaml.v2 v2.4.0
go: downloading github.com/projectdiscovery/sliceutil v0.0.0-20210804143453-61f3e7fd43ea
```

Once done, you can now run the tool. Help menu can be popped up to check the installation success

```
1. httpx --help
```

```
(root@kali)-[~]
# httpx --help
httpx is a fast and multi-purpose HTTP toolkit allow to run multiple probers using retryablehttp library.

Usage:
  httpx [flags]

Flags:
INPUT:
  -l, -list string      input file containing list of hosts to process
  -rr, -request string  file containing raw request

PROBES:
  -sc, -status-code      display response status-code
  -cl, -content-length  display response content-length
  -ct, -content-type    display response content-type
  -location              display response redirect location
  -favicon               display mmh3 hash for '/favicon.ico' file
  -hash string           display response body hash (supported: md5,mmh3,simhash,sha1,sha256,sha512)
  -rt, -response-time   display response time
  -lc, -line-count       display response body line count
  -wc, -word-count       display response body word count
  -title                 display page title
  -server, -web-server   display server name
  -td, -tech-detect      display technology in use based on wappalizer dataset
```

Basic Usage

Httpx tool accepts STDIN input for scanning. Here, we run a blank scan that only hits the server and does nothing and then the same scan with some basic options.

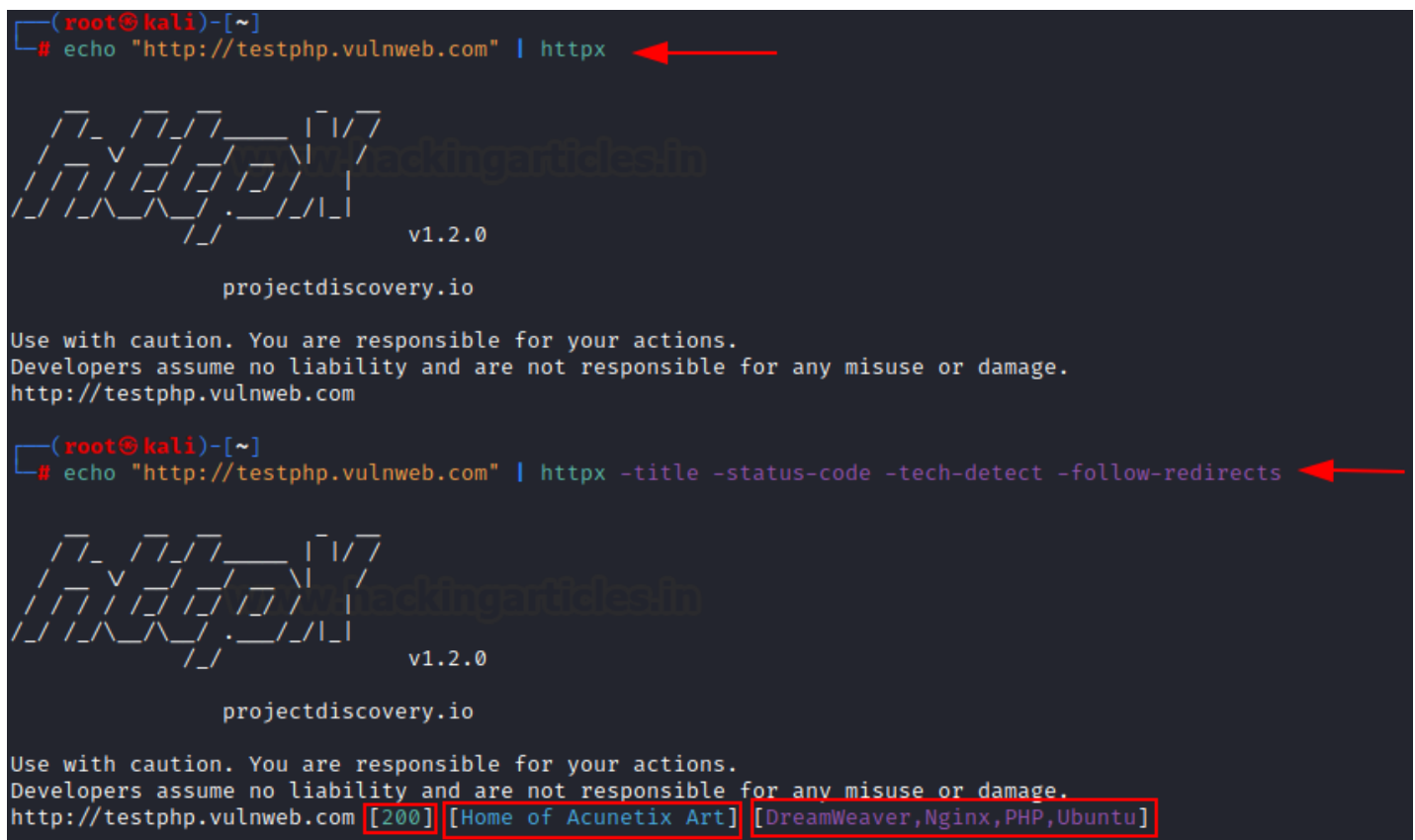
-title: displays the title of the webpage

-status-code: displays the response code. 200 being valid or OK status while 404 being the code for not found

-tech-detect: detects technology running behind the webpage

-follow-redirects: Enables following redirects and scans the following page too

1. `echo "http://testphp.vulnweb.com" | httpx`
2. `echo "http://testphp.vulnweb.com" | httpx -title -status-code -tech-detect -follow-redirects`



The screenshot shows a terminal window with two commands and their outputs. The first command is `echo "http://testphp.vulnweb.com" | httpx`, which outputs the version `v1.2.0` and the website `projectdiscovery.io`. The second command is `echo "http://testphp.vulnweb.com" | httpx -title -status-code -tech-detect -follow-redirects`, which outputs the same version and website, along with the status code `[200]`, the title `[Home of Acunetix Art]`, and the technologies `[DreamWeaver, Nginx, PHP, Ubuntu]`. Red arrows point to the command lines in both examples.

```
(root@kali)-[~]
# echo "http://testphp.vulnweb.com" | httpx

v1.2.0
projectdiscovery.io

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
http://testphp.vulnweb.com

(root@kali)-[~]
# echo "http://testphp.vulnweb.com" | httpx -title -status-code -tech-detect -follow-redirects

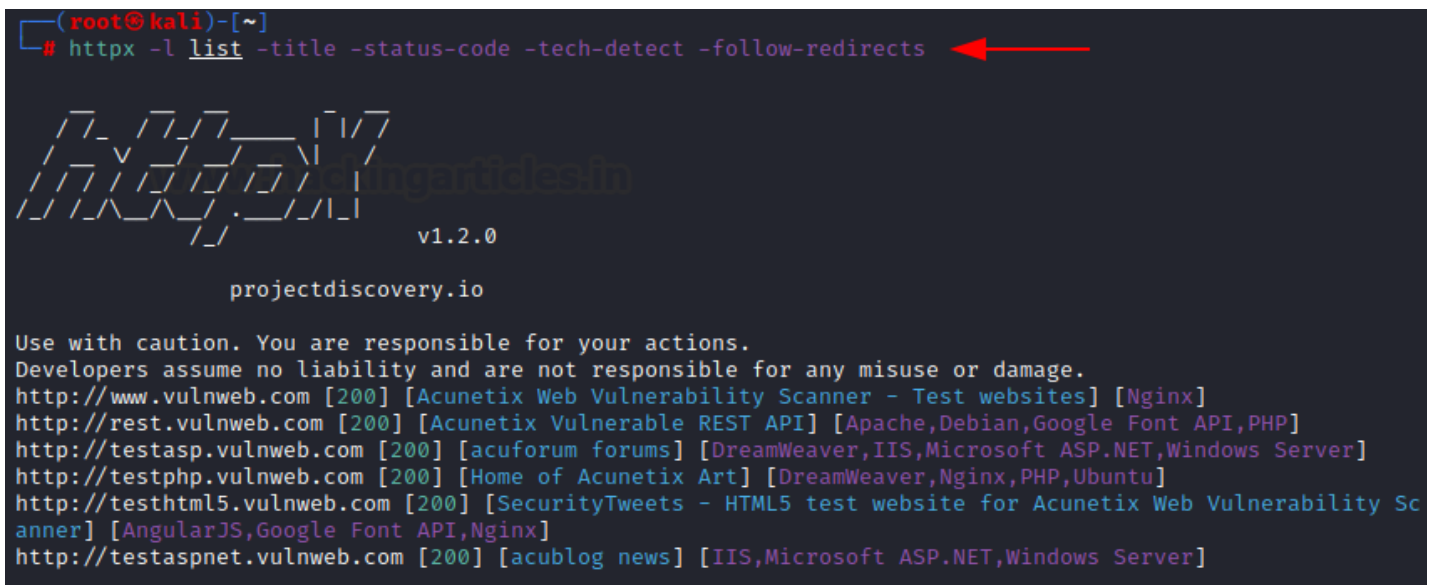
v1.2.0
projectdiscovery.io

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
http://testphp.vulnweb.com [200] [Home of Acunetix Art] [DreamWeaver, Nginx, PHP, Ubuntu]
```

The same can be run on a list of websites that can be fed to the tool using “-l” option

1. `httpx -l list -title -status-code -tech-detect -follow-redirects`

```
(root@kali)~# httpx -l list -title -status-code -tech-detect -follow-redirects
```



projectdiscovery.io

Use with caution. You are responsible for your actions.
 Developers assume no liability and are not responsible for any misuse or damage.

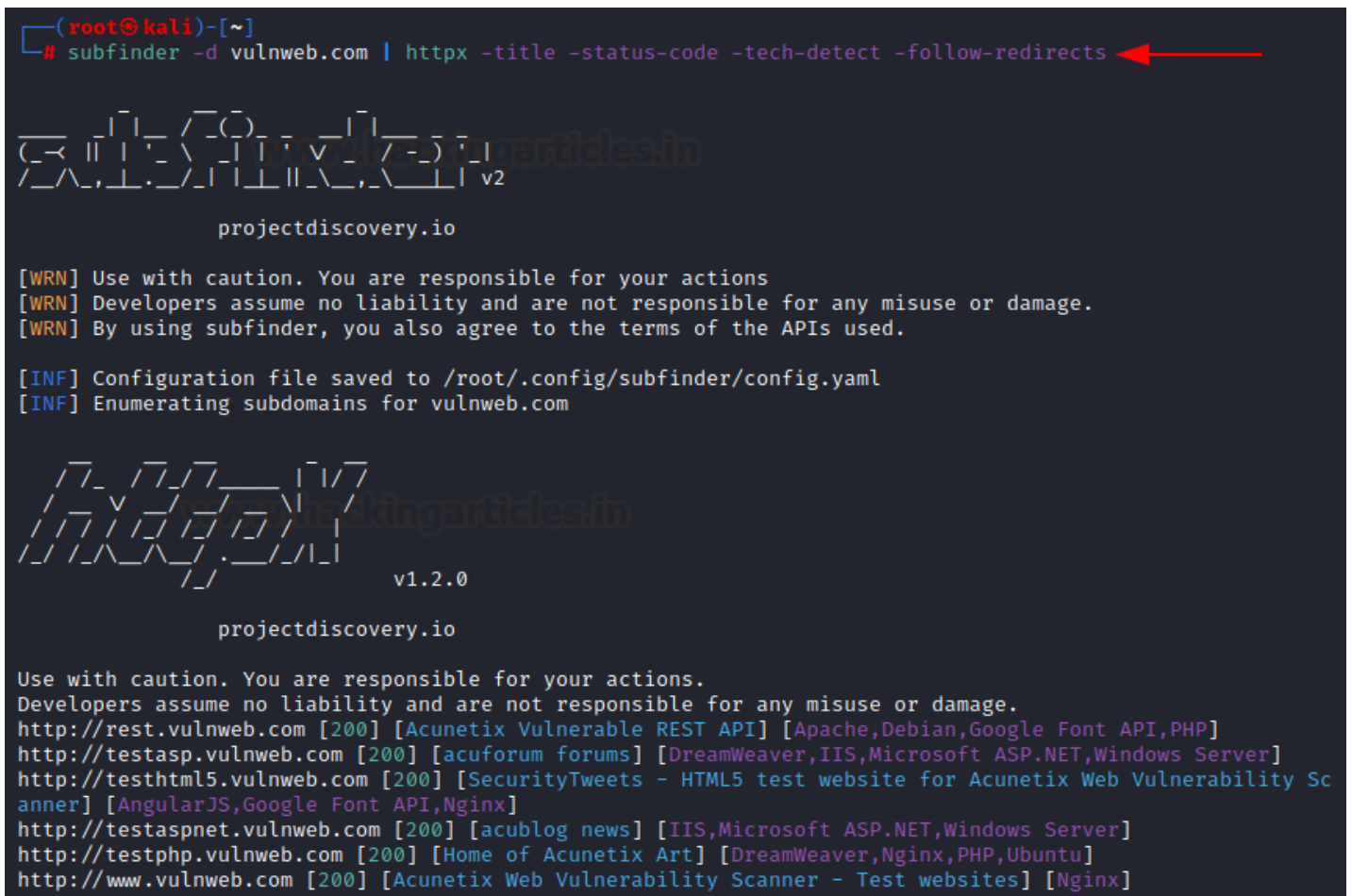
http://www.vulnweb.com [200] [Acunetix Web Vulnerability Scanner - Test websites] [Nginx]
 http://rest.vulnweb.com [200] [Acunetix Vulnerable REST API] [Apache,Debian,Google Font API,PHP]
 http://testasp.vulnweb.com [200] [acuforum forums] [DreamWeaver,IIS,Microsoft ASP.NET,Windows Server]
 http://testphp.vulnweb.com [200] [Home of Acunetix Art] [DreamWeaver,Nginx,PHP,Ubuntu]
 http://testhtml5.vulnweb.com [200] [SecurityTweets - HTML5 test website for Acunetix Web Vulnerability Scanner] [AngularJS,Google Font API,Nginx]
 http://testaspnet.vulnweb.com [200] [acublog news] [IIS,Microsoft ASP.NET,Windows Server]

Subdomain enum using subfinder and scan

Subfinder is another tool developed by projectdiscovery.io that enumerates and outputs subdomains. We can feed the STDOUT of subfinder to httpx and scan all the subdomains like so:

```
1. subfinder -d vulnweb.com | httpx -title -status-code -tech-detect -follow-redirects
```

```
(root@kali)~# subfinder -d vulnweb.com | httpx -title -status-code -tech-detect -follow-redirects
```



projectdiscovery.io

[WRN] Use with caution. You are responsible for your actions
 [WRN] Developers assume no liability and are not responsible for any misuse or damage.
 [WRN] By using subfinder, you also agree to the terms of the APIs used.

[INF] Configuration file saved to /root/.config/subfinder/config.yaml
 [INF] Enumerating subdomains for vulnweb.com

projectdiscovery.io

Use with caution. You are responsible for your actions.
 Developers assume no liability and are not responsible for any misuse or damage.

http://rest.vulnweb.com [200] [Acunetix Vulnerable REST API] [Apache,Debian,Google Font API,PHP]
 http://testasp.vulnweb.com [200] [acuforum forums] [DreamWeaver,IIS,Microsoft ASP.NET,Windows Server]
 http://testhtml5.vulnweb.com [200] [SecurityTweets - HTML5 test website for Acunetix Web Vulnerability Scanner] [AngularJS,Google Font API,Nginx]
 http://testaspnet.vulnweb.com [200] [acublog news] [IIS,Microsoft ASP.NET,Windows Server]
 http://testphp.vulnweb.com [200] [Home of Acunetix Art] [DreamWeaver,Nginx,PHP,Ubuntu]
 http://www.vulnweb.com [200] [Acunetix Web Vulnerability Scanner - Test websites] [Nginx]

Content probe

There are various modules that can refine how a response is rendered which is called a “probe.” These help us refine scan results. For example,

-sc: show HTTP response status code

-path: a specified path to check if it exists or not

```
1. httpx -l list -path /robots.txt -sc
```



```
(root@kali)-[~]
# httpx -l list -path /robots.txt -sc

projectdiscovery.io

v1.2.0

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
http://testasp.vulnweb.com/robots.txt [200]
http://testaspnet.vulnweb.com/robots.txt [200]
http://testhtml5.vulnweb.com/robots.txt [404]
http://www.vulnweb.com/robots.txt [404]
http://testphp.vulnweb.com/robots.txt [404]
http://rest.vulnweb.com/robots.txt [404]
```

httpx could be run using docker as well. Here, we feed a list of all subdomains as STDIN to httpx:

```
1. cat list | docker run -i projectdiscovery/httpx -title -status-code -tech-detect -follow-redirects
```



```
(root@kali)-[~]
# cat list | docker run -i projectdiscovery/httpx -title -status-code -tech-detect -follow-redirects

projectdiscovery.io

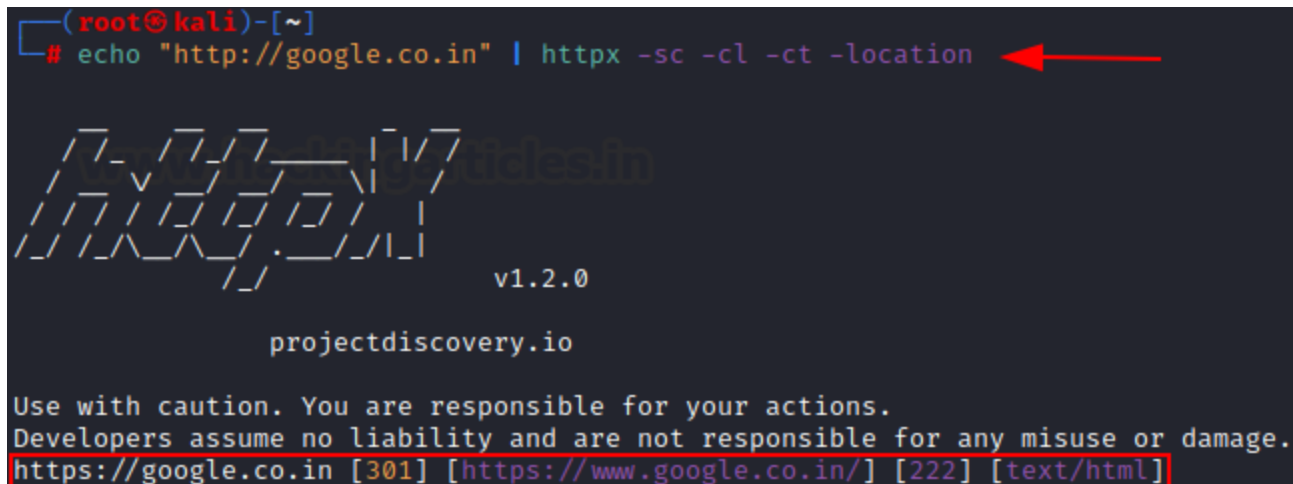
v1.2.0

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
http://www.vulnweb.com [200] [Acunetix Web Vulnerability Scanner - Test websites] [Nginx]
http://testasp.vulnweb.com [200] [acuforum forums] [DreamWeaver,IIS,Microsoft ASP.NET,Windows Server]
http://rest.vulnweb.com [200] [Acunetix Vulnerable REST API] [Apache,Debian,Google Font API,PHP]
http://testhtml5.vulnweb.com [200] [SecurityTweets - HTML5 test website for Acunetix Web Vulnerability Scanner] [A
Font API,Nginx]
http://testphp.vulnweb.com [200] [Home of Acunetix Art] [DreamWeaver,Nginx,PHP,Ubuntu]
http://testaspnet.vulnweb.com [200] [acublog news] [IIS,Microsoft ASP.NET,Windows Server]
```

There are various other probes that help us render better outputs

- location: website where redirected. Here, observe how http becomes https
- cl: displays the content length of the resulting web page
- ct: content type of the resulting web page. Mostly HTML

```
1. echo "http://google.co.in" | httpx -sc -cl -ct -location
```

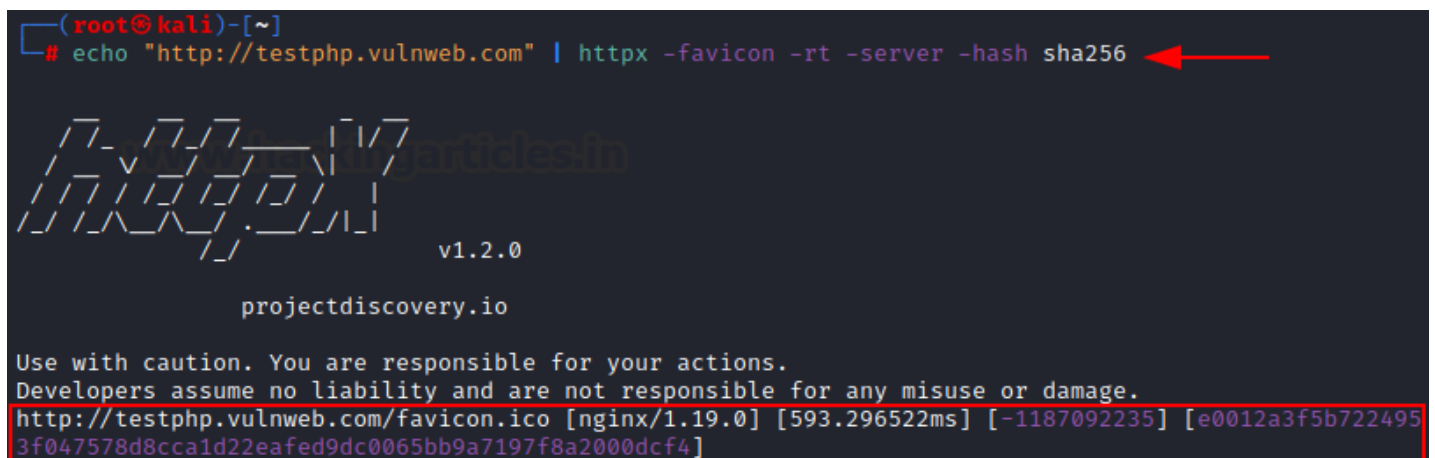


```
(root@kali)-[~]  
# echo "http://google.co.in" | httpx -sc -cl -ct -location  
  
v1.2.0  
projectdiscovery.io  
  
Use with caution. You are responsible for your actions.  
Developers assume no liability and are not responsible for any misuse or damage.  
https://google.co.in [301] [https://www.google.co.in/] [222] [text/html]
```

Some probes that are helpful for analysts and in-depth analysis

- favicon: fetches mmh3 hash of /favicon.ico file
- rt: shows the response time
- server: displays the server version and build
- hash: shows the webpage's content's hash

```
1. echo "http://testphp.vulnweb.com" | httpx -favicon -rt -server -hash sha256
```



```
(root@kali)-[~]  
# echo "http://testphp.vulnweb.com" | httpx -favicon -rt -server -hash sha256  
  
v1.2.0  
projectdiscovery.io  
  
Use with caution. You are responsible for your actions.  
Developers assume no liability and are not responsible for any misuse or damage.  
http://testphp.vulnweb.com/favicon.ico [nginx/1.19.0] [593.296522ms] [-1187092235] [e0012a3f5b7224953f047578d8cca1d22eafed9dc0065bb9a7197f8a2000dcf4]
```

- probe: displays the status of a single scan (success/failed)
- ip: displays the IP of the webserver

-cdn: displays the CDN/WAF if present

```
1. echo "https://shodan.io" | httpx -probe -ip -cdn
```

A terminal window with a dark background. The prompt is (root@kali)-[~]. The command # echo "https://shodan.io" | httpx -probe -ip -cdn is entered, with a red arrow pointing to the end of the command. The output shows a large ASCII art logo for 'hackingarticles.in' with 'v1.2.0' and 'projectdiscovery.io' below it. A disclaimer follows: 'Use with caution. You are responsible for your actions. Developers assume no liability and are not responsible for any misuse or damage.' The final output line is 'https://shodan.io [SUCCESS] [104.18.12.238] [cloudflare]'.

```
(root@kali)-[~]  
# echo "https://shodan.io" | httpx -probe -ip -cdn  
  
hackingarticles.in  
v1.2.0  
projectdiscovery.io  
  
Use with caution. You are responsible for your actions.  
Developers assume no liability and are not responsible for any misuse or damage.  
https://shodan.io [SUCCESS] [104.18.12.238] [cloudflare]
```

-lc: displays the line count of scanned web page

-wc: displays the word count of scanned web page

```
1. echo "http://testphp.vulnweb.com" | httpx -lc -wc
```

A terminal window with a dark background. The prompt is (root@kali)-[~]. The command # echo "http://testphp.vulnweb.com" | httpx -lc -wc is entered, with a red arrow pointing to the end of the command. The output shows the same ASCII art logo as the previous screenshot. The final output line is 'http://testphp.vulnweb.com [110] [514]'.

```
(root@kali)-[~]  
# echo "http://testphp.vulnweb.com" | httpx -lc -wc  
  
hackingarticles.in  
v1.2.0  
projectdiscovery.io  
  
Use with caution. You are responsible for your actions.  
Developers assume no liability and are not responsible for any misuse or damage.  
http://testphp.vulnweb.com [110] [514]
```

Content comparers

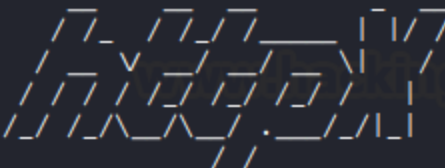
There are various comparers available in the tool that help us shortlist down an output. These are very helpful to trim down a list of unexpected output. For example,

-mc: matches the HTTP response code with the codes supplied in the list

```
1. cat list | httpx -mc 200,301,302 -sc
```

```
(root@kali)~[~]
# cat list | httpx -mc 200,301,302 -sc
```




 articles.in
 v1.2.0
 projectdiscovery.io

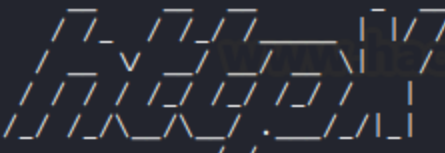
Use with caution. You are responsible for your actions.
 Developers assume no liability and are not responsible for any misuse or damage.
 http://www.vulnweb.com [200]
 http://testasp.vulnweb.com [200]
 http://testhtml5.vulnweb.com [200]
 http://testphp.vulnweb.com [200]
 http://rest.vulnweb.com [200]
 http://testaspnet.vulnweb.com [200]

-mlc: matches the line count with input provided

```
1. cat list | httpx -mlc 110 -lc
```

```
(root@kali)-[~]
# cat list | httpx -mhc 110 -lc
```




 projectdiscovery.io
 v1.2.0

Use with caution. You are responsible for your actions.
 Developers assume no liability and are not responsible for any misuse or damage.
<http://testphp.vulnweb.com> **[110]**

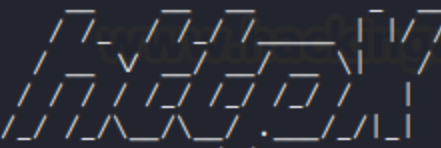
- cl: displays the content length of a webpage

-ml: matches the content length with the input provided and displays only the results matching the content length

```
1. cat list | httpx -ml 3563 -cl
```

```
(root@kali)-[~]
# cat list | httpx -ml 3563 -cl
```



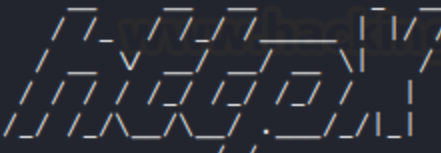

 v1.2.0
 projectdiscovery.io

Use with caution. You are responsible for your actions.
 Developers assume no liability and are not responsible for any misuse or damage.
<http://testasp.vulnweb.com> [3563]

-mwc: matches the word count and displays only the results with the same word count

```
1. cat list | httpx -mwc 580 -wc
```

```
(root@kali)-[~]
# cat list | httpx -mwc 580 -wc
```

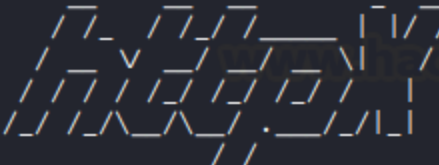

 articles.in
 v1.2.0
 projectdiscovery.io

Use with caution. You are responsible for your actions.
 Developers assume no liability and are not responsible for any misuse or damage.
<http://testaspnet.vulnweb.com> [580]

-ms: displays only the results where text on a page matches the provided string. Here, pages with “login” in their text are loaded

```
1. cat list | httpx -ms "login"
```

```
(root@kali)-[~]
# cat list | httpx -ms "login" ←
```


 v1.2.0
 projectdiscovery.io

Use with caution. You are responsible for your actions.
 Developers assume no liability and are not responsible for any misuse or damage.
<http://testphp.vulnweb.com>
<http://testasp.vulnweb.com>
<http://testaspnet.vulnweb.com>

-er: extract regular expressions. Displays only the results where the resulting pages match the regex pattern provided. An example regex is \w which compares the provided string with the resulting page's output.

```
1. echo "http://testphp.vulnweb.com" | httpx -er "\w test"
```



```
(root@kali)-[~]  
# echo "http://testphp.vulnweb.com" | httpx -er "\w test"   
  
www.hackingarticles.in  
v1.2.0  
projectdiscovery.io  
Use with caution. You are responsible for your actions.  
Developers assume no liability and are not responsible for any misuse or damage.  
http://testphp.vulnweb.com [u test,o test]
```

Here, you can see the output stands like u test, o test. The tool has filtered the following text and displayed it in output:



TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#)

search art

[Browse categories](#)

[Browse artists](#)

[Your cart](#)

[Signup](#)

[Your profile](#)

[Our guestbook](#)

[AJAX Demo](#)

Links

[Security art](#)

[PHP scanner](#)

[PHP vuln help](#)

[Fractal Explorer](#)

welcome to our page

Test site for Acunetix WVS.

[About Us](#) | [Privacy Policy](#) | [Contact Us](#) | [Shop](#) | [HTTP Parameter Pollution](#) | ©2019 Acunetix Ltd

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you **test** Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it **to test** other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

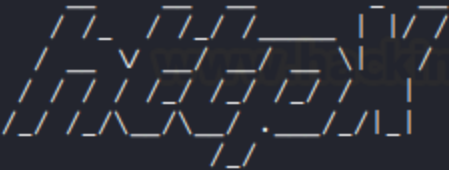
Content filters

Various filters are available at disposal in the tool that eliminates the results upon matching the criteria/condition provided. For example,

-fc: filters code. Tool only displays status codes not listed by fc (404 here so only 200 is visible)

1. cat list | httpx -sc
2. cat list | httpx -sc -fc 404


```
(root@kali)-[~]
# cat list | httpx -cl 
```



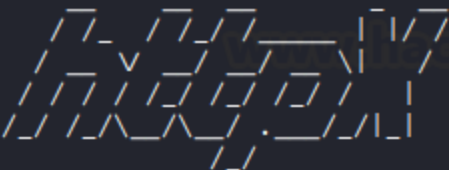
v1.2.0

projectdiscovery.io

Use with caution. You are responsible for your actions.
 Developers assume no liability and are not responsible for any misuse or damage.

http://www.vulnweb.com [4018]
 http://testasp.vulnweb.com [3563]
 http://testhtml5.vulnweb.com [6939]
 http://rest.vulnweb.com [3555]
 http://testphp.vulnweb.com/robots.php [16]
 http://testaspnet.vulnweb.com [12401]

```
(root@kali)-[~]
# cat list | httpx -cl -fl 16,12401 
```



v1.2.0

projectdiscovery.io

Use with caution. You are responsible for your actions.
 Developers assume no liability and are not responsible for any misuse or damage.

http://www.vulnweb.com [4018]
 http://rest.vulnweb.com [3555]
 http://testasp.vulnweb.com [3563]
 http://testhtml5.vulnweb.com [6939]

-fwc: filters the word count. Here, 3 and 580 is filtered so all the output except these two are visible

```
1. cat list | httpx -wc -fwc 3,580
```

```
(root@kali)-[~]
# cat list | httpx -wc

v1.2.0
projectdiscovery.io

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
http://www.vulnweb.com [482]
http://testhtml5.vulnweb.com [1483]
http://testasp.vulnweb.com [330]
http://rest.vulnweb.com [1397]
http://testphp.vulnweb.com/robots.php [3]
http://testaspnet.vulnweb.com [580]

(root@kali)-[~]
# cat list | httpx -wc -fwc 3,580

v1.2.0
projectdiscovery.io

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
http://www.vulnweb.com [482]
http://testasp.vulnweb.com [330]
http://testhtml5.vulnweb.com [1483]
http://rest.vulnweb.com [1397]
```

-flc: filter line count. Here, 2 and 89 is filtered so all the output except these two are visible

```
1. cat list | httpx -lc -flc 2,89
```



```
(root@kali)-[~]
# cat list | httpx -favicon

projectdiscovery.io

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
http://testasp.vulnweb.com/favicon.ico [1631801191]
http://www.vulnweb.com/favicon.ico [-1475802027]
http://testaspnet.vulnweb.com/favicon.ico [-1187092235]
http://rest.vulnweb.com/favicon.ico [-516366620]
http://testhtml5.vulnweb.com/favicon.ico [-1481944441]
http://testphp.vulnweb.com/robots.php/favicon.ico [-215994923]

(project@kali)-[~]
# cat list | httpx -favicon -ffc -215994923

projectdiscovery.io

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
http://www.vulnweb.com/favicon.ico [-1475802027]
http://testaspnet.vulnweb.com/favicon.ico [-1187092235]
http://testhtml5.vulnweb.com/favicon.ico [-1481944441]
http://testasp.vulnweb.com/favicon.ico [1631801191]
http://rest.vulnweb.com/favicon.ico [-516366620]
http://testphp.vulnweb.com/robots.php/favicon.ico [-1475802027]
```

Rates and Timeouts

There are various modules that let a user play around with the rate of scan and throttle the speed of the same. Some of these options are:

- t: specify the number of threads used for the scan. Can be as high as 150. Default 50.
- rl: specifies the rate limit in requests per second
- rlm: specifies the rate limit in requests per minute

```
1. cat list | httpx -sc -probe -t 10 -rl 1 -rlm 600
```

```
(root@kali)-[~]  
# cat list | httpx -sc -probe -t 10 -rl 1 -rlm 600  
  
v1.2.0  
projectdiscovery.io  
  
Use with caution. You are responsible for your actions.  
Developers assume no liability and are not responsible for any misuse or damage.  
http://random.vulnweb.com [FAILED]  
http://odincovo.vulnweb.com [FAILED]  
http://antivirus1.vulnweb.com [FAILED]  
http://ns1.vulnweb.com [FAILED]  
http://testaspnet.vulnweb.com [SUCCESS] [200]  
http://rest.vulnweb.com [SUCCESS] [200]  
http://testhtml5.vulnweb.com [SUCCESS] [200]  
http://testasp.vulnweb.com [SUCCESS] [200]  
http://a.vulnweb.com [FAILED]  
http://testphp.vulnweb.com/robots.php [SUCCESS] [404]  
http://tetphp.vulnweb.com [FAILED]  
http://virus.vulnweb.com [FAILED]  
http://viruswall.vulnweb.com [FAILED]  
http://www.virus.vulnweb.com [FAILED]  
http://www.test.php.vulnweb.com [FAILED]  
http://www.vulnweb.com [SUCCESS] [200]
```

-timeout: To abort the scan in specified seconds

-retries: Number of retries before aborting the scan

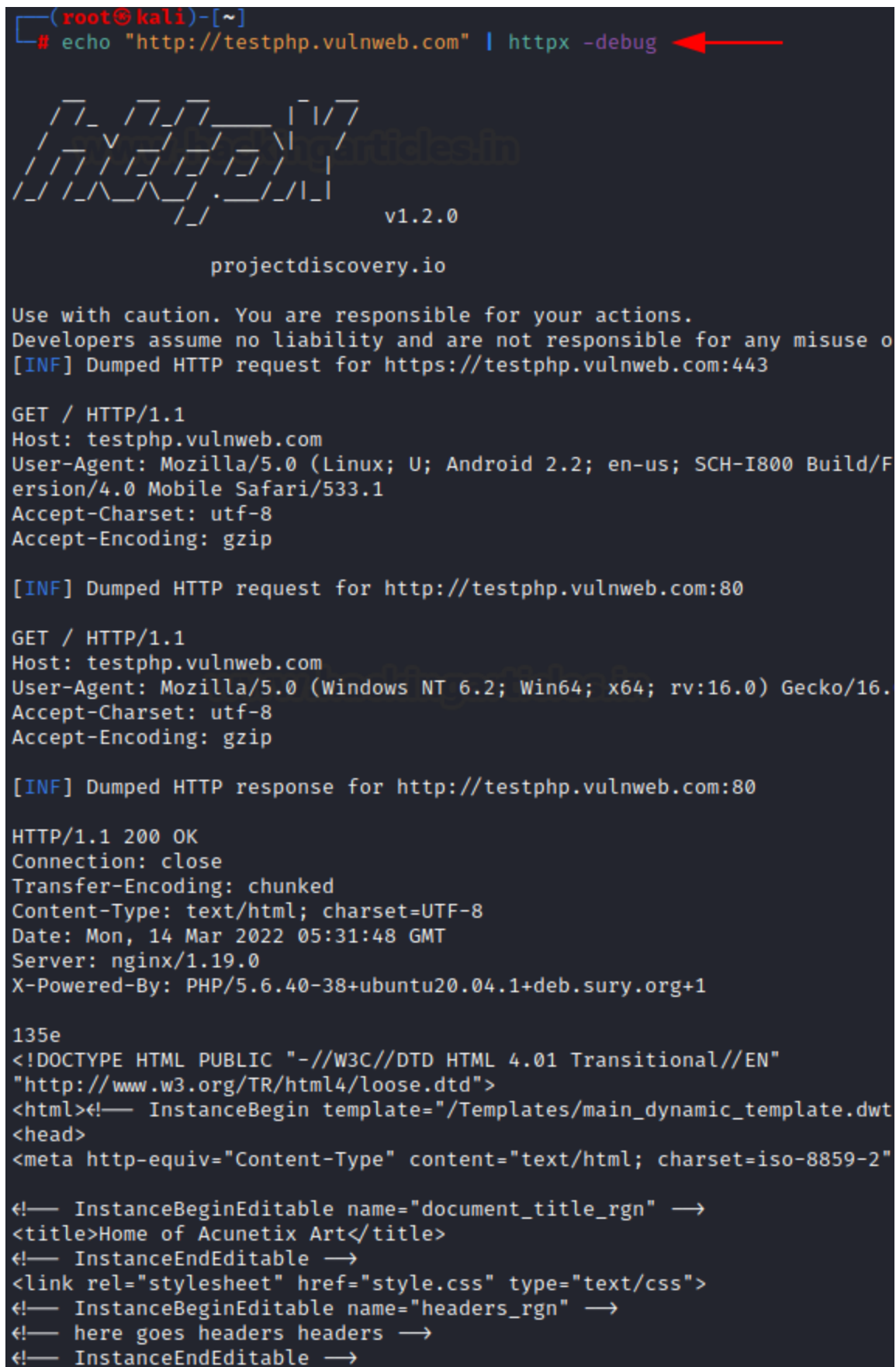
```
1. cat list | httpx -sc -probe -threads 50 -timeout 60 -retries 5
```



```

(root@kali)-[~]
# echo "http://testphp.vulnweb.com" | httpx -debug ←

```



Use with caution. You are responsible for your actions.
 Developers assume no liability and are not responsible for any misuse o
 [INF] Dumped HTTP request for https://testphp.vulnweb.com:443

```

GET / HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Linux; U; Android 2.2; en-us; SCH-I800 Build/F
ersion/4.0 Mobile Safari/533.1
Accept-Charset: utf-8
Accept-Encoding: gzip

```

[INF] Dumped HTTP request for http://testphp.vulnweb.com:80

```

GET / HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 6.2; Win64; x64; rv:16.0) Gecko/16.
Accept-Charset: utf-8
Accept-Encoding: gzip

```

[INF] Dumped HTTP response for http://testphp.vulnweb.com:80

```

HTTP/1.1 200 OK
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
Date: Mon, 14 Mar 2022 05:31:48 GMT
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1

```

```

135e
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><!-- InstanceBegin template="/Templates/main_dynamic_template.dwt
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2"

<!-- InstanceBeginEditable name="document_title_rgn" -->
<title>Home of Acunetix Art</title>
<!-- InstanceEndEditable -->
<link rel="stylesheet" href="style.css" type="text/css">
<!-- InstanceBeginEditable name="headers_rgn" -->
<!-- here goes headers headers -->
<!-- InstanceEndEditable -->

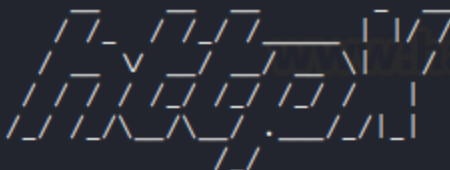
```

-debug-req: Displays the outgoing HTTP request

-debug-resp: Displays the corresponding HTTP response


```
(root@kali)-[~]
# echo "http://testphp.vulnweb.com" | httpx -probe -debug-req
```



 www.hackingarticles.in
v1.2.0
projectdiscovery.io

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
[INF] Dumped HTTP request for https://testphp.vulnweb.com:443

```
GET / HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Symbian/3; Series60/5.2 NokiaC7-00/012.003; Profile/MIDP-
WebKit/525 (KHTML, like Gecko) Version/3.0 BrowserNG/7.2.7.3 3gpp-gba
Accept-Charset: utf-8
Accept-Encoding: gzip
```

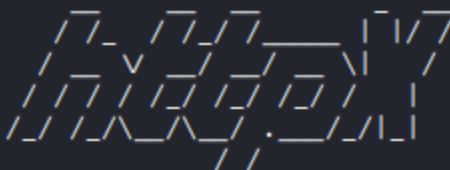
[INF] Dumped HTTP request for http://testphp.vulnweb.com:80

```
GET / HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Linux; Android 9; SM-A530W) AppleWebKit/537.36 (KHTML, li
ile Safari/537.36
Accept-Charset: utf-8
Accept-Encoding: gzip
```

http://testphp.vulnweb.com [SUCCESS]

```
(root@kali)-[~]
# echo "http://testphp.vulnweb.com" | httpx -probe -debug-req
```



 www.hackingarticles.in
v1.2.0
projectdiscovery.io

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
[INF] Dumped HTTP response for http://testphp.vulnweb.com:80

```
HTTP/1.1 200 OK
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
Date: Mon, 14 Mar 2022 05:33:28 GMT
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
```

135e
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

-stats: displays the current scan stats including completion percentage

```
1. cat list | httpx -stats
```



```
(root@kali)-[~]  
# cat list | httpx -stats  
  
v1.2.0  
projectdiscovery.io  
  
Use with caution. You are responsible for your actions.  
Developers assume no liability and are not responsible for any misuse or damage.  
[0:00:05] | RPS: 0 | Requests: 1 | Hosts: 16/16 (100%)  
http://rest.vulnweb.com  
http://www.vulnweb.com  
http://testhtml5.vulnweb.com  
http://testasp.vulnweb.com  
http://testphp.vulnweb.com/robots.php  
http://testaspnet.vulnweb.com
```

Filtering for SQL Injections

As we know that some types of SQL injections are reflected in the code output. We can detect such injections by filtering the output of a web page. In error-based SQLi, an error is thrown which is reflected in the output page. As you can see in the command below we have used -ms filter to compare and find such pages. Ideally, an attacker can give a list of input and find common SQLi vulnerabilities in a similar way. In the output below, where the vuln is found, httpx displays that website's name.

```
1. echo "http://testphp.vulnweb.com" | httpx -path "/listproducts.php?cat=1'" -ms "Error:  
You have an error in your SQL syntax;"
```

```
(root@kali)-[~]
# echo "http://testphp.vulnweb.com" | httpx -path "/listproducts.php?cat=1'" -ms "Error: You have
an error in your SQL syntax;"

projectdiscovery.io

v1.2.0

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
http://testphp.vulnweb.com/listproducts.php?cat=1'

(projectdiscovery.io)
# echo "http://testphp.vulnweb.com" | httpx -path "/listproducts.php?cat=1'" -ms "Error: You have
an error in your SQL syntax;"

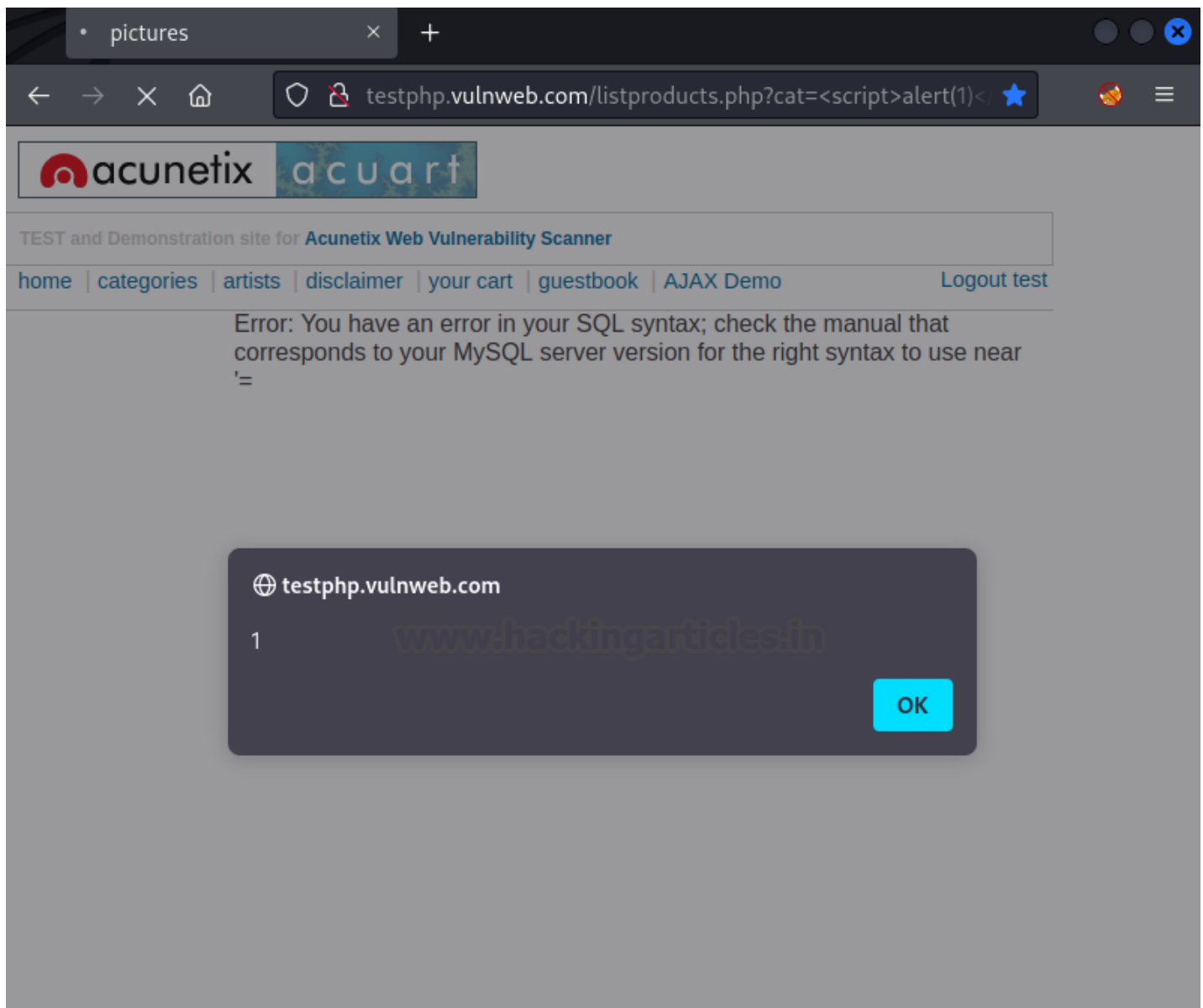
projectdiscovery.io

v1.2.0

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
```

Filtering for XSS reflections

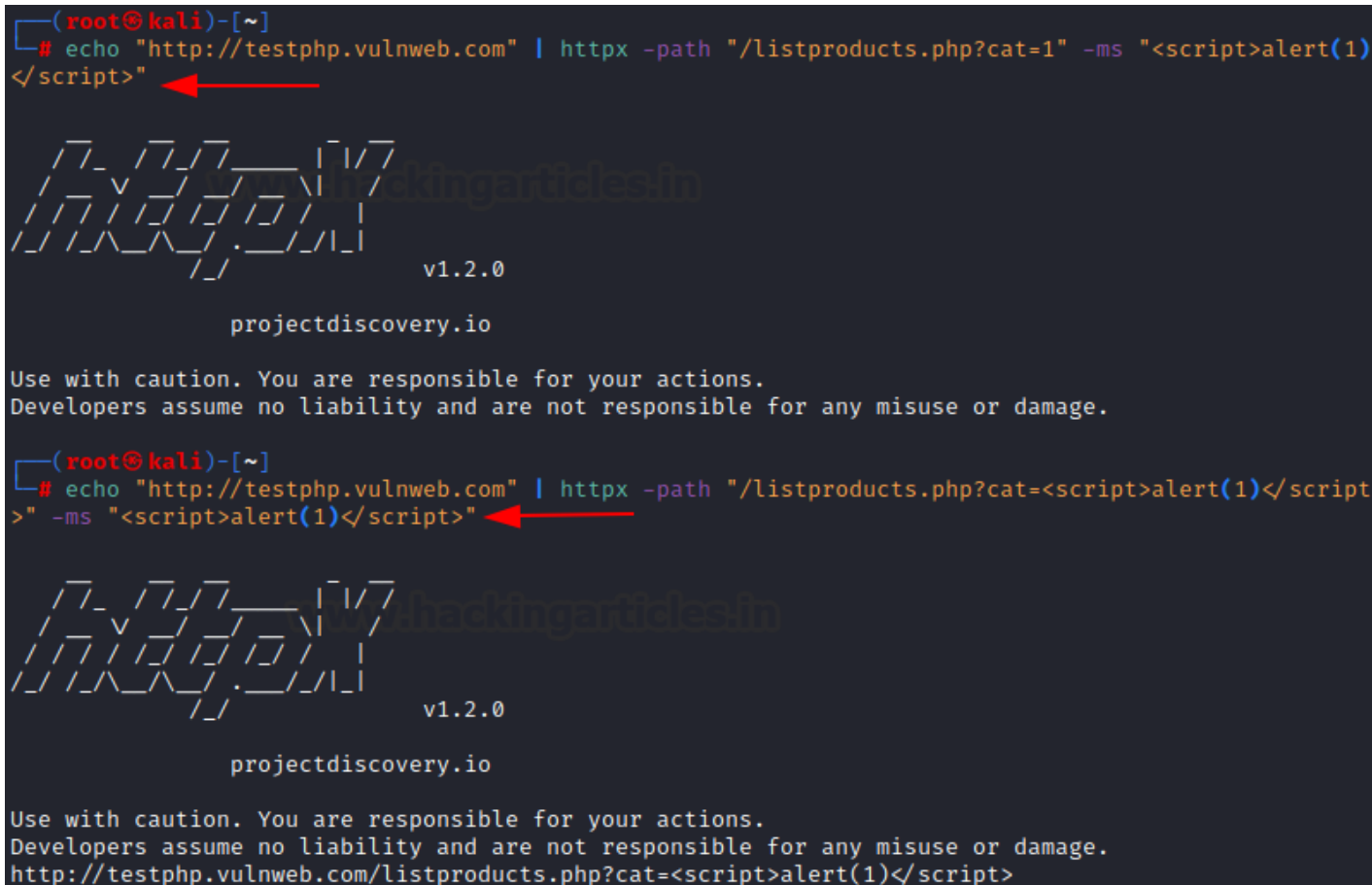
Reflected XSS by definition gets reflected in the web page's output.



An attacker can input a list of websites and then a list of path to check for reflected XSS in bunches. In the example below, the “-ms” module is used which is supposed to match the output webpage’s text content with the input provided. Since reflected XSS is shown in the output, the tool displays the name of the webpage where this vulnerability (payload output in the code) is observed.

```
1. echo "http://testphp.vulnweb.com" | httpx -path "/listproducts.php?cat=<script>alert(1)</script>" -ms "<script>alert(1)</script>"
```

```
(root@kali)-[~]
# echo "http://testphp.vulnweb.com" | httpx -path "/listproducts.php?cat=1" -ms "<script>alert(1)
</script>"
```



```
(root@kali)-[~]
# echo "http://testphp.vulnweb.com" | httpx -path "/listproducts.php?cat=<script>alert(1)</script>" -ms "<script>alert(1)</script>"
```

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
http://testphp.vulnweb.com/listproducts.php?cat=<script>alert(1)</script>

Web Page Fuzzing

Httpx is a great tool that can be used to fuzz web pages. “-path” module can be used to provide the name of the file to be fuzzed for existence on the server.

-path: path/list of paths to probe

```
1. echo "http://testphp.vulnweb.com" | httpx -probe -sc -path "/login.php"
```

```
(root@kali)-[~]
# echo "http://testphp.vulnweb.com" | httpx -probe -sc -path "/login.php"
```



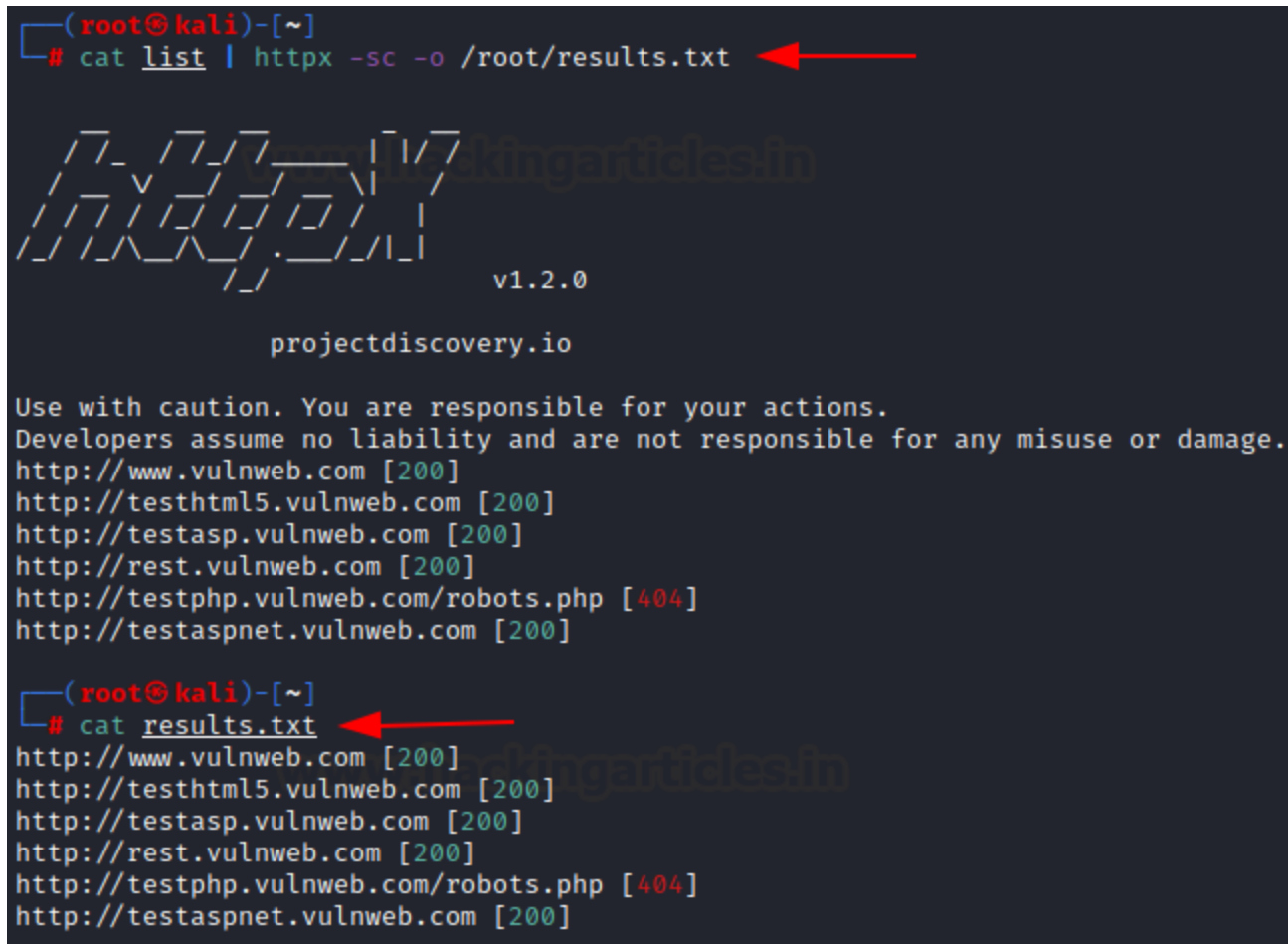
```
http://testphp.vulnweb.com/login.php [SUCCESS] [200]
```

File output

The scan results provided by the tool can also be exported for convenience. The most basic output is a text file with just webpages in every line. This can be useful for a variety of occasions while pentesting. Such modules are:

-o: Saves a result in a text output file

1. `cat list | httpx -sc -o /root/results.txt`
2. `cat results.txt`

A terminal window on a Kali Linux system. The prompt is (root@kali)-[~]. The first command is # cat list | httpx -sc -o /root/results.txt, with a red arrow pointing to it. The output shows a list of URLs and their status codes: http://www.vulnweb.com [200], http://testhtml5.vulnweb.com [200], http://testasp.vulnweb.com [200], http://rest.vulnweb.com [200], http://testphp.vulnweb.com/robots.php [404], and http://testaspnet.vulnweb.com [200]. Below this, the prompt is (root@kali)-[~] and the second command is # cat results.txt, with a red arrow pointing to it. The output is identical to the first command's output. A watermark 'hackingarticles.in' is visible in the background.

```
(root@kali)-[~]
# cat list | httpx -sc -o /root/results.txt

v1.2.0

projectdiscovery.io

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
http://www.vulnweb.com [200]
http://testhtml5.vulnweb.com [200]
http://testasp.vulnweb.com [200]
http://rest.vulnweb.com [200]
http://testphp.vulnweb.com/robots.php [404]
http://testaspnet.vulnweb.com [200]

(root@kali)-[~]
# cat results.txt
http://www.vulnweb.com [200]
http://testhtml5.vulnweb.com [200]
http://testasp.vulnweb.com [200]
http://rest.vulnweb.com [200]
http://testphp.vulnweb.com/robots.php [404]
http://testaspnet.vulnweb.com [200]
```

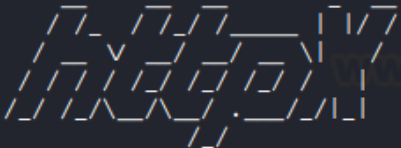
The same results can be saved in other formats too. Like,

-csv: Stores the scan results in CSV format. Default scan includes almost all of the content probes.

1. `cat list | httpx -sc -csv -o /root/results.csv`
2. `cat results.csv`

```
(root@kali)-[~]
# cat list | httpx -sc -csv -o /root/results.csv
```





 www.hackingarticles.in

 v1.2.0

 projectdiscovery.io

Use with caution. You are responsible for your actions.
 Developers assume no liability and are not responsible for any misuse or damage.

```
timestamp,request,response-header,scheme,port,path,a,cnames,url,input,location,title,error,webserver,response-body,content-type,method,host,content-length,chain-status-codes,status-code,tls-grab,csp,vhost,websocket,pipeline,http2,cdn,cdn-name,response-time,technologies,chain,final-url,failed,favicon-mmh3,hashes,lines,words
2022-03-14 02:19:00.594595206 -0400 EDT m=+5.610104495,,http,80,/,[44.228.249.3],[],http://www.vulnweb.com:80,www.vulnweb.com,,Acunetix Web Vulnerability Scanner - Test websites,,nginx/1.19.0,,text/html,GET,44.228.249.3,4018,[],200,<nil>,<nil>,false,false,false,false,false,,574.770308ms,[],[],false,,map[],74,482
2022-03-14 02:19:00.603454425 -0400 EDT m=+5.618963724,,http,80,/,[44.238.29.244],[],http://testasp.vulnweb.com:80,testasp.vulnweb.com,,acuforum forums,,Microsoft-IIS/8.5,,text/html,GET,44.238.29.244,3559,[],200,<nil>,<nil>,false,false,false,false,false,,577.646693ms,[],[],false,,map[],46,330
2022-03-14 02:19:00.615919637 -0400 EDT m=+5.631428916,,http,80,/,[44.228.249.3],[],http://testhtml5.vulnweb.com:80,testhtml5.vulnweb.com,,SecurityTweets - HTML5 test website for Acunetix Web Vulnerability Scanner,,nginx/1.19.0,,text/html,GET,44.228.249.3,6939,[],200,<nil>,<nil>,false,false,false,false,false,,575.104046ms,[],[],false,,map[],164,1483
2022-03-14 02:19:00.670953062 -0400 EDT m=+5.686462282,,http,80,/,[35.81.188.86],[],http://rest.vulnweb.com:80,rest.vulnweb.com,,Acunetix Vulnerable REST API,,Apache/2.4.25 (Debian),,text/html,GET,35.81.188.86,3555,[],200,<nil>,<nil>,false,false,false,false,false,,577.33823ms,[],[],false,,map[],138,1397
2022-03-14 02:19:00.732271541 -0400 EDT m=+5.747780760,,http,80,/robots.php,[],[],http://testphp.vulnweb.com:80/robots.php,testphp.vulnweb.com/robots.php,,,nginx/1.19.0,,text/html,GET,44.228.249.3,16,[],404,<nil>,<nil>,false,false,false,false,false,,577.537628ms,[],[],false,,map[],2,3
2022-03-14 02:19:00.886921593 -0400 EDT m=+5.902430822,,http,80,/,[44.238.29.244],[],http://testaspnet.vulnweb.com:80,testaspnet.vulnweb.com,,acublog news,,Microsoft-IIS/8.5,,text/html,GET,44.238.29.244,12401,[],200,<nil>,<nil>,false,false,false,false,false,,852.815467ms,[],[],false,,map[],89,580
```

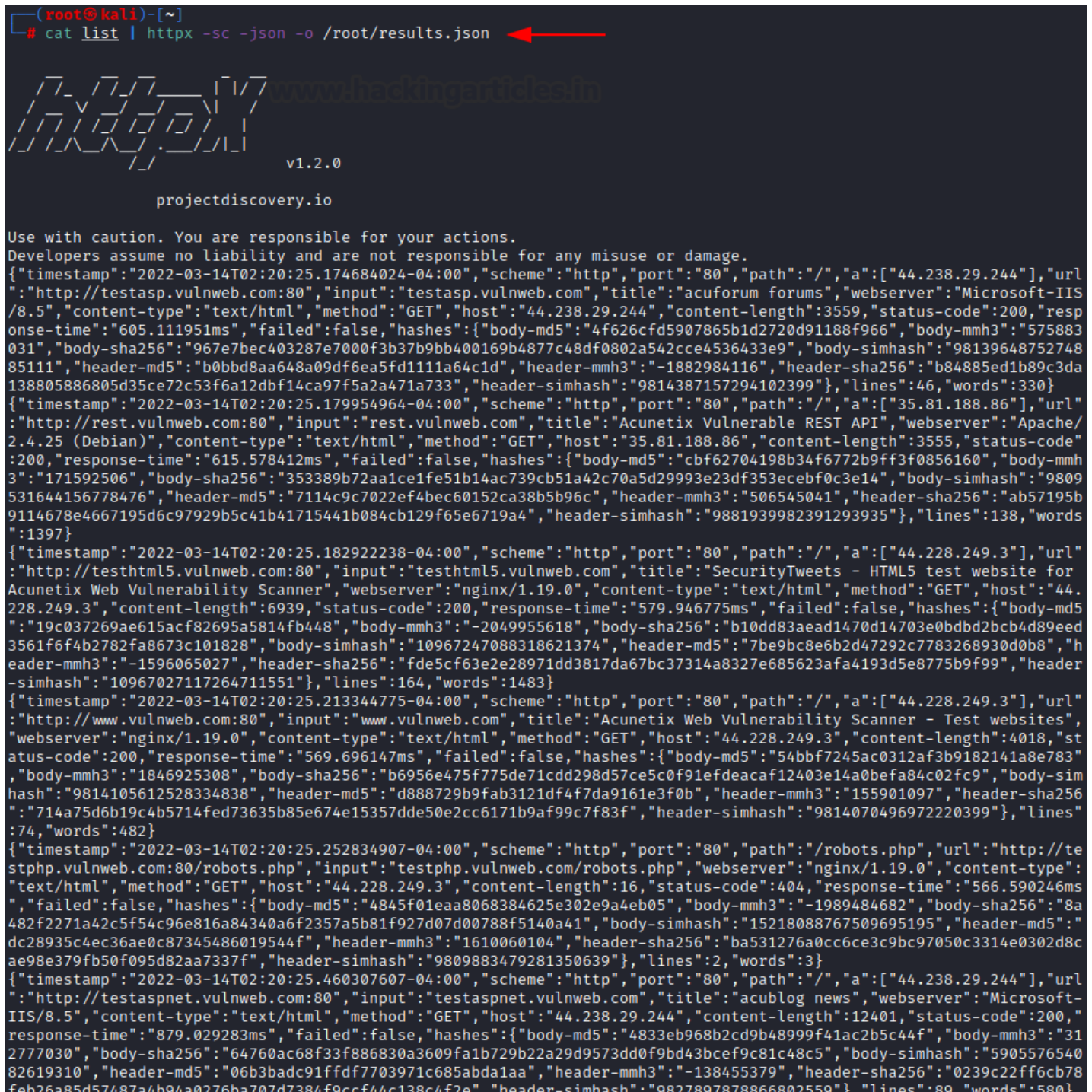
```
(root@kali)-[~]
# cat results.csv
timestamp,request,response-header,scheme,port,path,a,cnames,url,input,location,title,error,webserver,response-body,content-type,method,host,content-length,chain-status-codes,status-code,tls-grab,csp,vhost,websocket,pipeline,http2,cdn,cdn-name,response-time,technologies,chain,final-url,failed,favicon-mmh3,hashes,lines,words
2022-03-14 02:19:00.594595206 -0400 EDT m=+5.610104495,,http,80,/,[44.228.249.3],[],http://www.vulnweb.com:80,www.vulnweb.com,,Acunetix Web Vulnerability Scanner - Test websites,,nginx/1.19.0,,text/html,GET,44.228.249.3,4018,[],200,<nil>,<nil>,false,false,false,false,false,,574.770308ms,[],[],false,,map[],74,482
2022-03-14 02:19:00.603454425 -0400 EDT m=+5.618963724,,http,80,/,[44.238.29.244],[],http://testasp.vulnweb.com:80,testasp.vulnweb.com,,acuforum forums,,Microsoft-IIS/8.5,,text/html,GET,44.238.29.244,3559,[],200,<nil>,<nil>,false,false,false,false,false,,577.646693ms,[],[],false,,map[],46,330
2022-03-14 02:19:00.615919637 -0400 EDT m=+5.631428916,,http,80,/,[44.228.249.3],[],http://testhtml5.vulnweb.com:80,testhtml5.vulnweb.com,,SecurityTweets - HTML5 test website for Acunetix Web Vulnerability Scanner,,nginx/1.19.0,,text/html,GET,44.228.249.3,6939,[],200,<nil>,<nil>,false,false,false,false,false,,575.104046ms,[],[],false,,map[],164,1483
2022-03-14 02:19:00.670953062 -0400 EDT m=+5.686462282,,http,80,/,[35.81.188.86],[],http://rest.vulnweb.com:80,rest.vulnweb.com,,Acunetix Vulnerable REST API,,Apache/2.4.25 (Debian),,text/html,GET,35.81.188.86,3555,[],200,<nil>,<nil>,false,false,false,false,false,,577.33823ms,[],[],false,,map[],138,1397
```

-json: Stores the scan results in json format. Default scan includes almost all the content probes

```
1. cat list | httpx -sc -json -o /root/results.json
```



```
(root@kali)~[~]
# cat list | httpx -sc -json -o /root/results.json
```



Use with caution. You are responsible for your actions.
 Developers assume no liability and are not responsible for any misuse or damage.

```
{
  "timestamp": "2022-03-14T02:20:25.174684024-04:00",
  "scheme": "http",
  "port": "80",
  "path": "/",
  "a": ["44.238.29.244"],
  "url": "http://testasp.vulnweb.com:80",
  "input": "testasp.vulnweb.com",
  "title": "acuforum forums",
  "webserver": "Microsoft-IIS/8.5",
  "content-type": "text/html",
  "method": "GET",
  "host": "44.238.29.244",
  "content-length": 3559,
  "status-code": 200,
  "response-time": "605.111951ms",
  "failed": false,
  "hashes": {
    "body-md5": "4f626cfd5907865b1d2720d91188f966",
    "body-mmh3": "575883031",
    "body-sha256": "967e7bec403287e7000f3b37b9bb400169b4877c48df0802a542cce4536433e9",
    "body-simhash": "9813964875274885111",
    "header-md5": "b0bbd8aa648a09df6ea5fd111a64c1d",
    "header-mmh3": "-1882984116",
    "header-sha256": "b84885ed1b89c3da138805886805d35ce72c53f6a12dbf14ca97f5a2a471a733",
    "header-simhash": "9814387157294102399"
  },
  "lines": 46,
  "words": 330
}
{
  "timestamp": "2022-03-14T02:20:25.179954964-04:00",
  "scheme": "http",
  "port": "80",
  "path": "/",
  "a": ["35.81.188.86"],
  "url": "http://rest.vulnweb.com:80",
  "input": "rest.vulnweb.com",
  "title": "Acunetix Vulnerable REST API",
  "webserver": "Apache/2.4.25 (Debian)",
  "content-type": "text/html",
  "method": "GET",
  "host": "35.81.188.86",
  "content-length": 3555,
  "status-code": 200,
  "response-time": "615.578412ms",
  "failed": false,
  "hashes": {
    "body-md5": "cbf62704198b34f6772b9ff3f0856160",
    "body-mmh3": "171592506",
    "body-sha256": "353389b72aa1ce1fe51b14ac739cb51a42c70a5d29993e23df353cebf0c3e14",
    "body-simhash": "9809531644156778476",
    "header-md5": "7114c9c7022ef4bec60152ca38b5b96c",
    "header-mmh3": "506545041",
    "header-sha256": "ab57195b9114678e4667195d6c97929b5c41b41715441b084cb129f65e6719a4",
    "header-simhash": "9881939982391293935"
  },
  "lines": 138,
  "words": 1397
}
{
  "timestamp": "2022-03-14T02:20:25.182922238-04:00",
  "scheme": "http",
  "port": "80",
  "path": "/",
  "a": ["44.228.249.3"],
  "url": "http://testhtml5.vulnweb.com:80",
  "input": "testhtml5.vulnweb.com",
  "title": "SecurityTweets - HTML5 test website for Acunetix Web Vulnerability Scanner",
  "webserver": "nginx/1.19.0",
  "content-type": "text/html",
  "method": "GET",
  "host": "44.228.249.3",
  "content-length": 6939,
  "status-code": 200,
  "response-time": "579.946775ms",
  "failed": false,
  "hashes": {
    "body-md5": "19c037269ae615acf82695a5814fb448",
    "body-mmh3": "-2049955618",
    "body-sha256": "b10dd83aead1470d14703e0dbdb2bcb4d89eed3561f6f4b2782fa8673c101828",
    "body-simhash": "10967247088318621374",
    "header-md5": "7be9bc8e6b2d47292c7783268930d0b8",
    "header-mmh3": "-1596065027",
    "header-sha256": "fde5cf63e2e28971dd3817da67bc37314a8327e685623afa4193d5e8775b9f99",
    "header-simhash": "10967027117264711551"
  },
  "lines": 164,
  "words": 1483
}
{
  "timestamp": "2022-03-14T02:20:25.213344775-04:00",
  "scheme": "http",
  "port": "80",
  "path": "/",
  "a": ["44.228.249.3"],
  "url": "http://www.vulnweb.com:80",
  "input": "www.vulnweb.com",
  "title": "Acunetix Web Vulnerability Scanner - Test websites",
  "webserver": "nginx/1.19.0",
  "content-type": "text/html",
  "method": "GET",
  "host": "44.228.249.3",
  "content-length": 4018,
  "status-code": 200,
  "response-time": "569.696147ms",
  "failed": false,
  "hashes": {
    "body-md5": "54bbf7245ac0312af3b9182141a8e783",
    "body-mmh3": "1846925308",
    "body-sha256": "b6956e475f775de71cdd298d57ce5c0f91efdeacaf12403e14a0bfa84c02fc9",
    "body-simhash": "9814105612528334838",
    "header-md5": "d888729b9fab3121df4f7da9161e3f0b",
    "header-mmh3": "155901097",
    "header-sha256": "714a75d6b19c4b5714fed73635b85e674e15357dde50e2cc6171b9af99c7f83f",
    "header-simhash": "9814070496972220399"
  },
  "lines": 74,
  "words": 482
}
{
  "timestamp": "2022-03-14T02:20:25.252834907-04:00",
  "scheme": "http",
  "port": "80",
  "path": "/robots.php",
  "url": "http://testphp.vulnweb.com:80/robots.php",
  "input": "testphp.vulnweb.com/robots.php",
  "title": "testphp.vulnweb.com/robots.php",
  "webserver": "nginx/1.19.0",
  "content-type": "text/html",
  "method": "GET",
  "host": "44.228.249.3",
  "content-length": 16,
  "status-code": 404,
  "response-time": "566.590246ms",
  "failed": false,
  "hashes": {
    "body-md5": "4845f01eaa8068384625e302e9a4eb05",
    "body-mmh3": "-1989484682",
    "body-sha256": "8a482f2271a42c5f54c96e816a84340a6f2357a5b81f927d07d00788f5140a41",
    "body-simhash": "15218088767509695195",
    "header-md5": "dc28935c4ec36ae0c87345486019544f",
    "header-mmh3": "1610060104",
    "header-sha256": "ba531276a0cc6c3c9bc97050c3314e0302d8cae98e379fb50f095d82aa7337f",
    "header-simhash": "9809883479281350639"
  },
  "lines": 2,
  "words": 3
}
{
  "timestamp": "2022-03-14T02:20:25.460307607-04:00",
  "scheme": "http",
  "port": "80",
  "path": "/",
  "a": ["44.238.29.244"],
  "url": "http://testaspnet.vulnweb.com:80",
  "input": "testaspnet.vulnweb.com",
  "title": "acublog news",
  "webserver": "Microsoft-IIS/8.5",
  "content-type": "text/html",
  "method": "GET",
  "host": "44.238.29.244",
  "content-length": 12401,
  "status-code": 200,
  "response-time": "879.029283ms",
  "failed": false,
  "hashes": {
    "body-md5": "4833eb968b2cd9b48999f41ac2b5c44f",
    "body-mmh3": "312777030",
    "body-sha256": "64760ac68f33f886830a3609fa1b729b22a29d9573dd0f9bd43bcef9c81c48c5",
    "body-simhash": "590557654082619310",
    "header-md5": "06b3badc91ffdf7703971c685abda1aa",
    "header-mmh3": "-138455379",
    "header-sha256": "0239c22ff6cb78feb26a85d57487a4b94a0276ba707d384f9ccf44c138c4f2e",
    "header-simhash": "9827897878866802559"
  },
  "lines": 89,
  "words": 580
}
```

-srd: stores corresponding HTTP responses in custom directory with naming: "URL.txt"

1. `cat list | httpx -sc -o /root/results.txt -srd /root/responses`
2. `cat /root/responses/rest.vulnweb.com.txt`


```
(root@kali)~# mkdir responses

(root@kali)~# cat list | httpx -sc -o /root/results.txt -srd /root/responses

projectdiscovery.io v1.2.0

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
http://www.vulnweb.com [200]
http://testasp.vulnweb.com [200]
http://testhtml5.vulnweb.com [200]
http://rest.vulnweb.com [200]
http://testphp.vulnweb.com/robots.php [404]
http://testaspnet.vulnweb.com [200]

(root@kali)~# cd responses/

(root@kali)~/responses# ls
rest.vulnweb.com.txt      testasp.vulnweb.com.txt  testphp.vulnweb.com_robots.php.txt
testaspnet.vulnweb.com.txt testhtml5.vulnweb.com.txt www.vulnweb.com.txt

(root@kali)~/responses# cat rest.vulnweb.com.txt
HTTP/1.1 200 OK
Connection: close
Content-Type: text/html; charset=UTF-8
Date: Mon, 14 Mar 2022 06:25:57 GMT
Server: Apache/2.4.25 (Debian)
Vary: Accept-Encoding
X-Powered-By: PHP/7.1.26

<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="author" content="Acunetix">

  <title>Acunetix Vulnerable REST API</title>

  <!-- Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Nunito:200,600" rel="stylesheet">

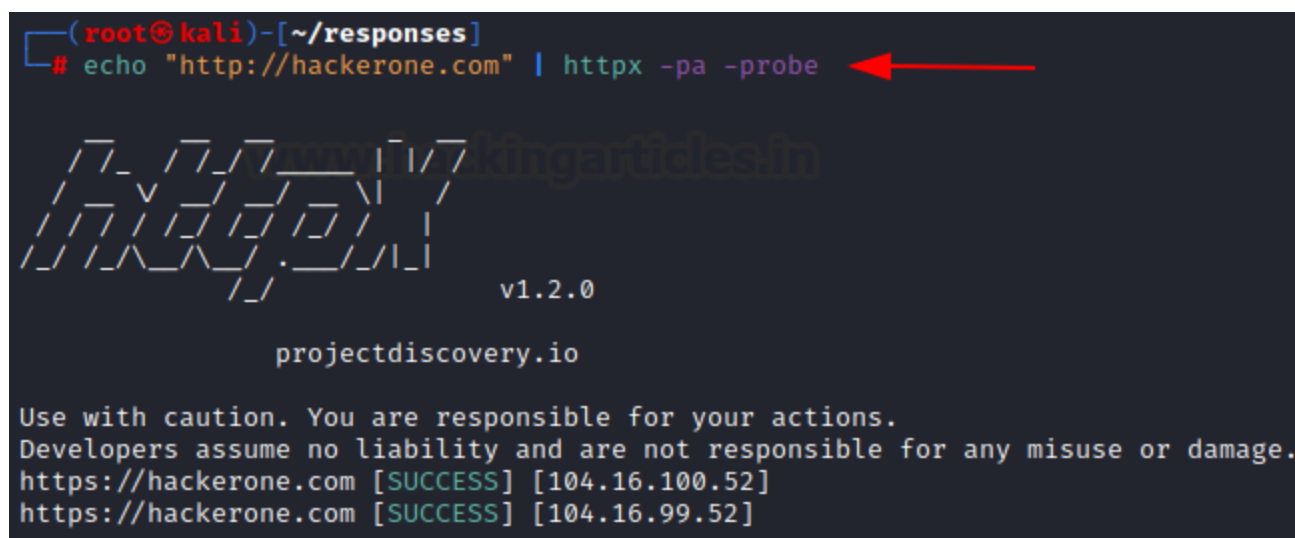
  <!-- Styles -->
  <style>
    html,
    body {
```

TCP/IP customizations

Some filters are available to conduct an in-depth reconnaissance. These filters are extremely helpful in cases where an attacker needs to conduct basic network-level reconnaissance too.

-pa: probes all IPs associated with the same host provided. Often the same website is utilizing multiple IP addresses for different purposes.

```
1. echo "http://hackerone.com" | httpx -pa -probe
```



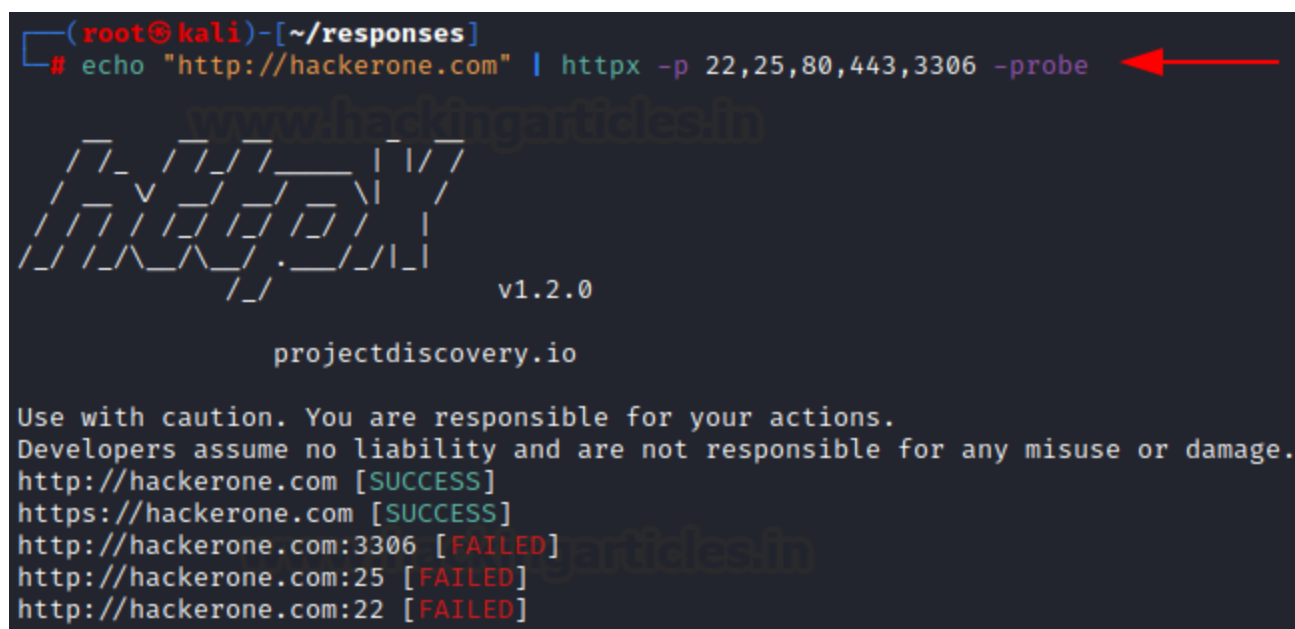
```
(root@kali)-[~/responses]
# echo "http://hackerone.com" | httpx -pa -probe

www.hackingarticles.in
v1.2.0
projectdiscovery.io

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
https://hackerone.com [SUCCESS] [104.16.100.52]
https://hackerone.com [SUCCESS] [104.16.99.52]
```

-p: scans the specified ports either as a list (in the format 80,443) or by providing absolute range (format 1-1023)

```
1. echo "http://hackerone.com" | httpx -p 22,25,80,443,3306 -probe
```



```
(root@kali)-[~/responses]
# echo "http://hackerone.com" | httpx -p 22,25,80,443,3306 -probe

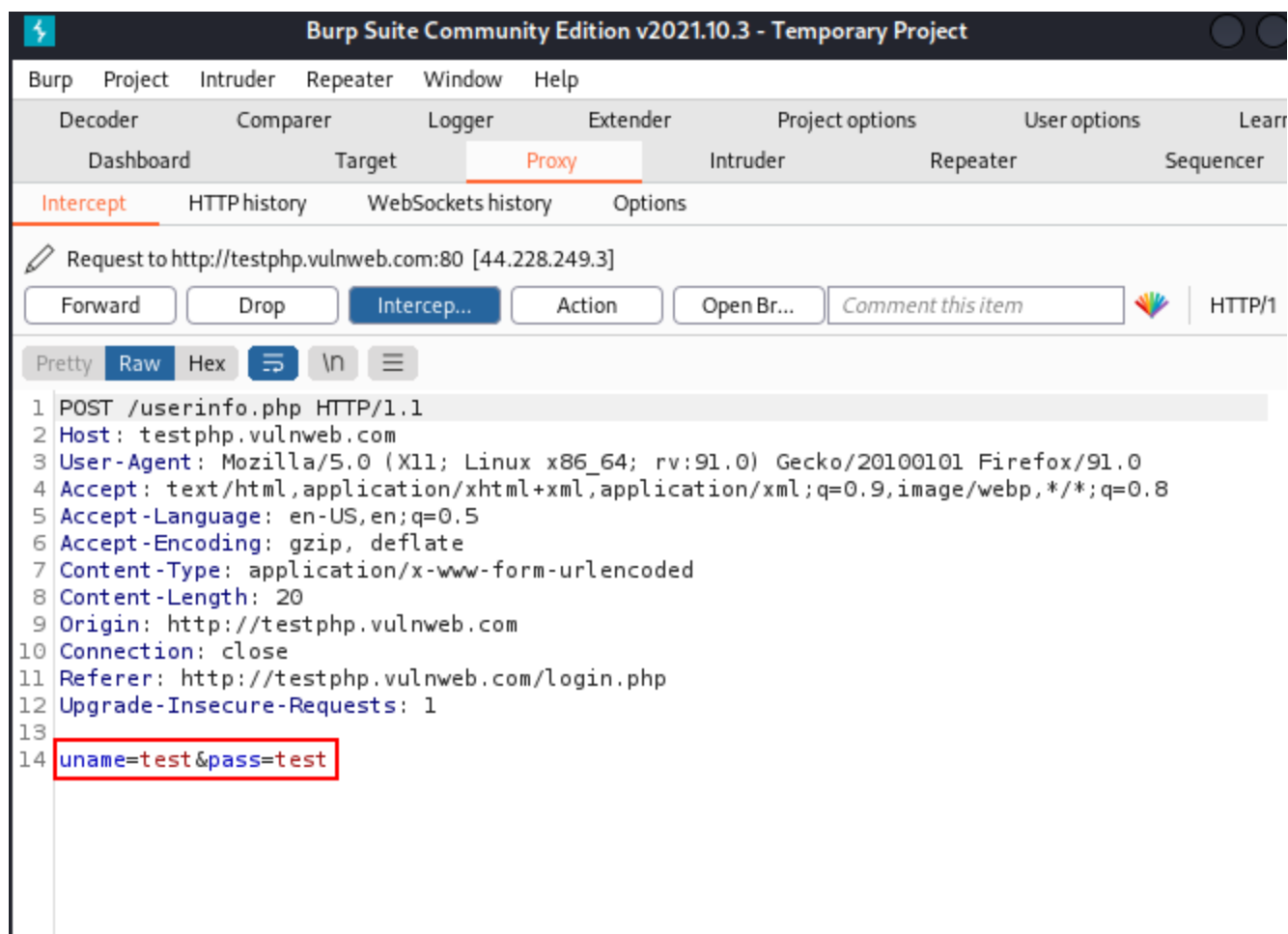
www.hackingarticles.in
v1.2.0
projectdiscovery.io

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
http://hackerone.com [SUCCESS]
https://hackerone.com [SUCCESS]
http://hackerone.com:3306 [FAILED]
http://hackerone.com:25 [FAILED]
http://hackerone.com:22 [FAILED]
```

POST Login

Httpx can also be used to send POST requests. It can also be used to log into a page and read responses. For example, the page /userinfo.php is a login portal and can be logged in with

credentials test:test. The corresponding request in the burp suite looks like



To replicate the same request, httpx provides various modules

-x: specify the HTTP request options. GET, POST, PUT etc.

-H: provides custom headers to be sent

-body: specifies the additional data in the body to be sent along with the request

As you can see in the screenshot below, the tool has logged in (200 OK) and displayed the output of the profile page.

```
1. echo "http://testphp.vulnweb.com" | httpx -debug-resp -x post -path "/userinfo.php" -H
  "Cookie: login=test%2Ftest" -body "uname=test&pass=test"
```

```
(root@kali)-[~]
# echo "http://testphp.vulnweb.com" | httpx -debug-resp -x post -path "/userinfo.php" -H "Cookie: login=test%2Ftest" -body "uname=test&pass=test"
v1.2.0
projectdiscovery.io

Use with caution. You are responsible for your actions.
Developers assume no liability and are not responsible for any misuse or damage.
[INF] Dumped HTTP response for http://testphp.vulnweb.com:80/userinfo.php
HTTP/1.1 200 OK
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
Date: Mon, 14 Mar 2022 07:07:38 GMT
Server: nginx/1.19.0
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1

179f
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" codeOutsideHTMIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">

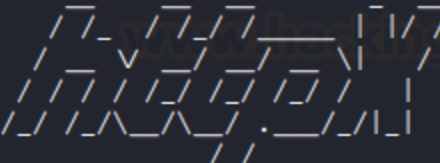
<!-- InstanceBeginEditable name="document_title_rgn" -->
<title>user info</title>
<!-- InstanceEndEditable -->
<link rel="stylesheet" href="style.css" type="text/css">
<!-- InstanceBeginEditable name="headers_rgn" -->
<!-- here goes headers headers -->
<!-- InstanceEndEditable -->
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_reloadPage(init) { //reloads the window if Nav4 resized
  if (init==true) with (navigator) {if ((appName=="Netscape")&&(parseInt(appVersion)=4)) {
    document.MM_pgW=innerWidth; document.MM_pgH=innerHeight; onresize=MM_reloadPage; }}
```

HTTP Methods Probe

The “-x all” option probes all the HTTP OPTIONS (request methods) and displays which options are permitted on the webpage. It is a nifty tool for pentesting. As it is visible, all the options are permitted on the webserver.

```
1. echo "http://testphp.vulnweb.com" | httpx -x all -probe
```

```
(root@kali)~# echo "http://testphp.vulnweb.com" | httpx -x all -probe
```



 v1.2.0

 projectdiscovery.io

Use with caution. You are responsible for your actions.
 Developers assume no liability and are not responsible for any misuse or damage.

```

http://testphp.vulnweb.com [SUCCESS] [POST]
http://testphp.vulnweb.com [SUCCESS] [OPTIONS]
http://testphp.vulnweb.com [SUCCESS] [PUT]
http://testphp.vulnweb.com [SUCCESS] [CONNECT]
http://testphp.vulnweb.com [SUCCESS] [HEAD]
http://testphp.vulnweb.com [SUCCESS] [GET]
http://testphp.vulnweb.com [SUCCESS] [DELETE]
http://testphp.vulnweb.com [SUCCESS] [TRACE]
http://testphp.vulnweb.com [SUCCESS] [PATCH]
  
```


Join Our Training Program





Categories

Cryptography & Steganography

CTF Challenges

Cyber Forensics

Database Hacking

Footprinting

Hacking Tools

Kali Linux

Nmap

Others

Password Cracking

Penetration Testing

Pentest Lab Setup

Privilege Escalation

Red Teaming

Social Engineering Toolkit

Uncategorized

Website Hacking

Window Password Hacking

Wireless Hacking

Wireless Penetration Testing

Archives

Select Month



You may like

Burpsuite for Pentester: Authorize

January 22, 2024

Burpsuite for Pentester: Logger++

October 30, 2023